

As per claim 1, Thorne discloses a method for creating a self destructing document, comprising the steps of creating an executable module which instructs a computer to automatically delete the document to which the executable module is attached when the document, based on a preselected expiration date is expired; attaching the executable module to the document [such as an Email software with the security options including a notice indicating an impending software will be self-destructed, the capability of sender to control the ability of recipient to copy, forward, print and store document, user selects the class of security which it is desired to impose by attach to the document, Thorne col 1 lines 42-col 2 line 55, col 6 lines 22-67, col 7 lines 1-42, col 8 lines 27-42]. Therefore Thorne provides all means necessary to a skilled [sic] in the art to create an Email

Thorne also teaches the Private message with the security features such as automatically deleted document after being accessed by the recipient or after giving a time limit or other predetermined events (print, forward, copy, store), notified and user given a warning and option when attempt to process the message as a design choice [Thorne col 8 lines 59-67, col 10 lines 1-62, col 11 lines 5-53, col 12 lines 10-16]. However Thorne fails to explicitly teach the executable code is attached to the email (or document). Beck discloses a Email message with attachment, and encryption and decryption keys, automatically deleted by a time limit [Beck col 7 lines 1-18]. Therefore it would have been obvious to one having ordinary skill in the art at the time the invention was made to modify the email with attachment taught by Beck with Thorne's system. By doing so it would improve the security and reliability on the data processing network.

Claim 1 recites:

1. A method for creating a self-destructing document, comprising the steps of:
creating an executable module which instructs a computer to automatically delete the document to which the executable module is attached when the document, based upon a preselected expiration date, is expired;
attaching the executable module to the document.

As claim 6 contains similar limitations to claim 1, except that claim pertains to an e-mail messaging system, the rejection of claim 6 will be discussed together with the rejection of claim

1. Claim 6 recites :

6. A self-destructing e-mail messaging system, comprising:
an executable module, the executable module configured to instruct a computer to automatically delete a message to which the executable module is attached when the message, based upon a preselected expiration date, is expired;

an e-mail messaging system, the e-mail messaging system configured to create the message and to transmit the message, the e-mail messaging system attaching the executable module to the message prior to transmission.

At the outset, applicants wish to point out to the Examiner that with respect to claims 1-10, 13-15 and 17, Thorne is not prior art because the filing date for the reference is July 17, 1997, one full month after the filing date of June 17, 1997 for the parent provisional application. Despite the fact that Thorne is not a valid prior art reference, applicants will address the Examiner's arguments with respect to Thorne-Beck .

It is respectfully submitted that the Thorne patent does not disclose or suggest creating an executable module that automatically deletes a document or e-mail message, or attaching such a module to a document or e-mail message as recited in claims 1 and 6. The Thorne patent purports to disclose an e-mail system that destroys e-mail messages based on a date. The information specifying how and when to destroy the e-mail (security parameters) is stored as data fields in the header of the e-mail itself, not in a separate, attached executable module. There is no mention anywhere in the reference of creating an executable module, much less attaching such a module to a document, or any other type of computer file.

In support of his position, the Examiner points to the text "...an Email software with security options including a notice indicating an impending software will be self-destructed..." from column 2, line 17 of the Thorne patent. This text is taken from the "Background" section of the Thorne patent and is referring to an entirely different patent. The entire paragraph from which that quote is taken states:

U.S. Pat. No. 5,014,234 to Edwards, Jr., entitled System With Software Usage Timer and Counter for Allowing Limited Use but Preventing Continued Unauthorized Use of Protected Software, issued May 7, 1991, pertains to prevention of continued unauthorized use of protected software. Copies of software are installed for a limited time. If the software is not registered within the time limit, the software is disabled. The system provides a notice indicating an impending software "Self-Destruct" in the event of a failure to register.

There is nothing in this section that can lead anyone to the method of creating a self-

destructing document as recited in claim 1 or the e-mail messaging system of claim 6. Again, the Examiner refers to a section that does not teach an attached executable module. Furthermore, the only mention of self-destruct functionality is made indirectly by talking about a warning notice of self-destruction. There is no discussion of how self-destruction would occur or how it could possibly operate.

The Examiner also states that the Thorne system includes the "...capability of sender to control the ability of recipient to copy, forward, print and store document, user selects the class of security which it is desired to impose by attach to the document..." These are not elements of claim 1 and the applicants fail to see why the Examiner has included them in his rejection of claim 1. In any event, at column 6, lines 22-67, cited by the Examiner, there is a list of security parameters that can be accorded to an e-mail allowing "...the sender to control the ability of the recipient..." to manipulate the e-mail in various ways. Column 7, line 21 provides the basis for the Examiner's argument that the user "...selects the class of security which it is desired to impose by attach to the document..." In it, Thorne states "...the user selects the class of security which it is desired to impose. This selection may control the maximum duration of life which the user is able to attach to the document." The word "attach" is used in the sense that a certain property is given to the document, not that an executable module, or any functional element for that matter, is attached. It seems that when the Examiner states that the user "...selects the class of security which it is desired to impose by attach to the document..." he is under the impression that Thorne attaches something to the document. This is simply incorrect, Thorne does not attach anything to a document.

The Examiner looks to column 8 lines 27-42 to support his argument that Thorne teaches the limitations of claim 1. The pertinent part of that section, starting at line 27, states:

The E-mail application in use according to the invention responds to the setting of the fields of the template by causing the packet assembler to insert into the message packet header flags to cause each recipient computer or processor to respond to the commands created by completion of the template.

The e-mail system of Thorne responds to the "setting of fields" by inserting the data field

values into the header of an e-mail via the packet assembler. These values, or “header flags” are read by the recipient computer to delete the e-mail in accordance with the header flags. There is no attached executable module and no self-destruct of the e-mail. Destruction of the data in an e-mail is accomplished by the e-mail system, not by the e-mail, or any part of the e-mail itself.

The Examiner looks to Beck because Thorne “...fails to explicitly teach [sic] the executable code is attached to the email (or document). Beck discloses a Email message with attachment, and encryption and decryption keys, automatically deleted by a time limit [Beck col 7 lines 1-18].”

Beck does nothing to bolster the weakness of Thorne. Beck purports to disclose an e-mail system with an alternate method of sending e-mail attachments. Instead of sending the attachment itself, a reference to the network address of the attachment is attached to the e-mail. The recipient accesses the attachment by clicking on its reference, causing the recipient computer to retrieve the attachment from the network address. There is no attached executable, and the only discussion of deletion after a condition is met relates to deleting an e-mail attachment, not the e-mail itself. Beck merely provides a way to efficiently use restricted bandwidth and storage capacities by not sending entire files as attachments with an e-mail. The purpose and function of Beck is entirely different and unrelated to that of the present invention.

In any event, the portion of Beck cited by the Examiner states, in pertinent part, “...an attachment may...be automatically deleted...after a given time limit...” The attachment Beck refers to, however, is not even the e-mail attachment itself, it is a reference to the network address of the e-mail attachment. In its abstract, Beck states “An attachment reference is generated, comprising the network address of the attachment. The attachment reference is transmitted from the sender to the at least one recipient.” The e-mail itself remains untouched, even after the expiration of the time limit. So Beck could not possibly teach, as the Examiner states, “a Email message...automatically deleted by a time limit.”

Aside from Thorne being an invalid reference, the combination of Thorne and Beck does not render the elements of claims 1 and 6 obvious because of the differences outlined above.

Further, because Thorne deals with e-mail security and Beck deals with efficiently utilizing limited bandwidth and storage capacity, there is no motivation to combine the references. Moreover, the combination of Thorne and Beck results in an e-mail system that attaches network references to e-mails and stores security parameters in the e-mail's header. There is still no method for creating a self-destructing document or an e-mail messaging system with an attached executable for automatically deleting an expired e-mail.

For the foregoing reasons, it is respectfully submitted that the Examiner's rejection of claims 1 and 6 is overcome and should be withdrawn.

Claims 2-5 and 18, and claims 7-10, 13-15 and 17 depend from and include the limitations of claims 1 and 6, respectively. Withdrawal of the Examiner's rejection of these claims is therefore requested on this basis as well. However, as these claims also include additional limitations which further distinguish these claims over the Thorne-Beck combination, applicants will address the specific points raised by the Examiner with regard to these dependent claims.

With respect to claims 2-4, the Examiner states "...Thorne-Beck discloses the executable module as an executable code, program, macro as program software including Email application [Thorne col 6 lines 22-30]."

As previously discussed, neither Thorne nor Beck disclose or suggest an executable module, so they cannot, alone or in combination, disclose "...the executable module as an executable code, program, macro..." The section of Thorne cited by the Examiner does not help his position because it does not talk about codes or programs, only data storage memory and means of data transmission. The section states:

Thus the program software, including E-mail applications, may reside at different times on a variety of media, including the various memories, disk drives and other storage media. The computer platform also may receive software in the form of carrier wave signals bearing digital code, via various communication ports and interfaces, such as the data interface(s) 253, 255 and the modem 254.

The above section of Thorne is referring to the different types of memory storage that the “program software, including Email applications” may reside on. This is clear from the remainder of the sentence which states that the “...software...may reside...on a variety of media, including...memories, disk drives and other storage media.” Thorne is talking about the different physical devices on which software may reside, such as RAM memory (various memories), hard disk drives and floppy disk drives (disk drives) and CD-ROM (other storage media). Thorne is not talking about the form the software program, or a set of computer instructions, may take, namely an executable code, executable program or macro as recited in claims 2-4.

Beck also lacks any disclosure or suggestion of an executable module. In Beck, an Internet address is attached to an e-mail. This is not, and cannot be considered an executable code, executable program or macro as recited in claims 2-4. Therefore, withdrawal of the rejection of claims 2-4 is requested on this basis as well.

It is the Examiner’s opinion, with respect to claim 5, that “...Thorne-Beck disclose the step of executing the executable module when the document is opened as a design choice of program software [Thorne col 1 lines 42-57, col 6 lines 22-67].”

Applicants reiterate that Thorne-Beck cannot possibly disclose the step of creating an executable module if it does not first disclose or suggest an executable module, which it does not. Therefore, Thorne-Beck does not disclose the steps of claim 5.

The section of Thorne cited by the Examiner to support his rejection of claim 5, column 1, lines 42-57 merely summarizes U.S. Patent Nos. 4,899,299 and 5,051,891 to MacPhail. The paragraph states, in pertinent part:

...The originator of a document specifies an ownership expiration date, and the enterprise operating the system establishes an expiration date. An algorithm causes deletion of a document from storage when a particular relationship exists among the current date and the two expiration dates. For example, the system deletes a message of the current date if later than both of the expiration dates.

Nowhere in the above section is an executable module mentioned, much less an executable module attached to a document. In addition, the step of opening an e-mail is not even eluded to, so the above cited paragraph cannot serve as the basis for rejecting claim 5 which recites executing an executable module when an e-mail to which the module is attached is opened.

Column 6, lines 22-67, also cited to support the rejection of claim 5, is discussed above and only mention the various types of storage media available for computer data and lists a number of functions the Thorne system claims to provide. An executable module cannot be found in this portion of Thorne either. Accordingly, "...the step of executing the executable module when the document is opened..." cannot be in the cited passage.

Beck does not add anything to Thorne as Beck does not have, anywhere in the reference, any teaching or suggestion of an executable module, much less attaching an executable module to an e-mail. Beck only attaches an Internet address reference to an e-mail, so there cannot be an attached executable module that executes every time an e-mail to which the executable is attached is opened.

Because the references do not, alone or in combination, recite or even suggest the limitations of claim 5, withdrawal of the rejection of claim 5 is respectfully requested on that basis as well as on its dependency from an allowable base claim.

It is the Examiner's opinion that claims 7-10, 13-15 and 17-19 "contain the same limitations that were addressed in rejecting claims 1-5 above." Using the same rationale, claims 7-10, 13-15 and 17-19 are rejected.

Claims 7-9 contain similar subject matter of claims 2-4 and depend from claim 6, which is believed to be allowable. Therefore, based on their dependency from an allowable base claim and on the differences set out in the arguments for claims 2-4, which apply here as well, the withdrawal of the rejection of claims 7-9 is respectfully requested.

Claim 10 depends from and includes all the limitations of claim 6 and further recites “...wherein the executable module is configured to overwrite the message with null characters.” By using the same rationale for rejecting claims 1-5 to reject claim 10, the Examiner has not provided grounds for the rejection of claim 10 because the subject matter of claim 10 cannot be found in claims 1-5. Claim 1 recites automatically deleting a document, but deleting is not the same as overwriting a message with null characters. Therefore, no grounds for the rejection of claim 10 are provided. In any event, claim 10 should be allowable based on its dependency from claim 6, an allowable base claim. Accordingly, the rejection of claim 10 is overcome and should be withdrawn.

Claims 13-15 depend from and include all the limitations of claim 6. Accordingly, claims 13-15 should be allowable based on their dependency therefrom.

Claims 17 recites:

17. A self-destructing e-mail messaging system, comprising:
 - an executable module, the executable module configured to instruct a computer to automatically delete an e-mail message to which the executable module is attached when a predetermined condition is met, wherein said predetermined condition is selected from the group consisting of an attempt to print the message, an attempt to copy the message and an attempt to forward the message;
 - an e-mail messaging system, the e-mail messaging system configured to create the message and to transmit the message, the e-mail messaging system attaching the executable module to the message prior to transmission.

With regard to the limitations in claim 17 that are common to claim 6, namely an executable module configured to instruct a computer to automatically delete an e-mail message to which the executable module is attached when a predetermined condition is met, and an e-mail messaging system configured to create the message and to transmit the message, the e-mail messaging system attaching the executable module to the message prior to transmission, the arguments set forth above for those elements of claim 6 apply here as well with respect to claim 17. Withdrawal of the Examiner’s rejection is requested on this basis as well.

Claim 18 depends from claim 1 and further recites "...the document is an encrypted document, and wherein the executable module is configured to instruct the computer to decrypt the document if the document is not expired, and to delete the document if the document is expired." These limitations cannot be found anywhere in claims 1-5. Encryption is not a part of claims 1-5 and is never mentioned. Basing the rejection of claim 18 on the rejection of claims 1-5 is therefore flawed as the Examiner has not pointed out where the limitations of claim 18 can be found in the prior art.

The same is true of claim 19, which depends from claim 6 and further recites the same limitations except that claim pertains to a message instead of a document. By basing the rejection of claim 19 on the rationale for rejecting claims 1-5, the Examiner has failed to show how the prior art teaches the limitations of claim 19 because he never addresses an encrypted message and the executable module configured to instruct the computer to decrypt the message if the message is not expired, and to delete the message if the message is expired.

Therefore, for the reasons set forth above, in addition to their dependency from an allowable base claim, the rejection of claims 18 and 19 are overcome and should be withdrawn.

B. Claims 20-47

Claim 20 recites:

20. A method for creating a virtual container containing a digital object, comprising the steps of :

 creating a virtual container, the virtual container residing in contiguous locations in an electronic storage media of a computer, the virtual container including a header portion and a digital object portion;

 selecting a digital object for insertion into the virtual container;

 applying an encryption technique to the digital object to create an encrypted digital object;

 writing the encrypted digital object into the digital object portion;

 selecting an expiration date for the digital object;

 writing information indicative of the expiration date into the header portion of the virtual container.

To support his rejection of claim 20, the Examiner relies on the rationale used to reject claims 1-5, stating that claim 20 contains the same limitations as those claims. This simply is not correct. Claim 20 teaches a method for creating a virtual container in contiguous memory locations for an encrypted digital object where the virtual container stores an expiration date for the digital object. Claim 1 recites a method for creating a self-destructing document by creating an executable module that automatically deletes a document to which it is attached when the document expires and attaching the executable module to the document. Applicants fail to see how claim 20 can be rejected on the same grounds as claims 1-5.

The virtual container created by the method of claim 20 is not the same as the executable module resulting from claim 1. The virtual container resides in contiguous memory locations in a computer with a header portion and a digital object portion. A digital object goes into the digital object portion and information about an expiration date for the digital object goes into the header portion. The executable module of claim 1 attaches to a document; residing in contiguous memory locations and having a header portion and a digital object portion are not limitations of claim 1. The executable module is a set of computer instructions that, as stated in claim 1, automatically delete a document to which the executable module is attached when the document expires. These limitations cannot be found in claim 20.

Accordingly, the Examiner has provided no indication of how these features of claim 20 are disclosed or suggested by the applied references. The Examiner has failed to establish a *prima facie* case of obviousness and provide adequate grounds for rejection because he has not found each and every limitation of claim 20 taught or suggested by the prior art. MPEP § 2143.3. Withdrawal of the Examiner's rejection of claim 20 is therefore respectfully requested.

Even if the Examiner had tried to argue that the limitations of claim 20 can be found in Thorne and Beck, he would be mistaken. A method for creating a virtual container residing in contiguous locations in an electronic storage media of a computer, the virtual container including a header portion and a digital object portion cannot be found in Thorne or Beck, alone or in combination. It follows that without the first step of the claimed method, the steps that follow in the claim cannot possibly be found in the cited references. There is not even anything that

resembles the claimed virtual container anywhere in the references. Withdrawal of the Examiner's rejection is requested on this basis as well.

Similarly, claim 21 recites a method for extracting a document from a virtual container. Claim 22 recites a virtual container system. Claim 23, a method for creating a virtual container and extracting a digital object from a virtual container, and claim 32, a method for transmitting a destructible document. The Examiner has not shown where any of the limitations from any of claims 21-23 and 32 can be found in the prior art. A prima facie case of obviousness has not been made for these claims as well.

The rejection of claims 21-23 and 32 are overcome and should be withdrawn for the same reasons that the rejection of claim 20 is overcome. The Examiner has not indicated where the elements of claims 21-23 and 32 can be found in the references, the virtual container of claims 21-23 and 32 are nowhere to be found in the Thorne and Beck, alone or in combination.

As claims 24-27 and 34 depend from and incorporate the limitations of claim 20, withdrawal of the Examiner's rejection of these claims is also requested on the same bases for which the withdrawal of the rejection of claim 20 is requested. Namely, the lack of a prima facie case, lack of the claim elements in the cited references, alone or in combination, and for using the same rationale to reject claims 1-5. The same reasoning applies to claims 28-31, 35, 38 and 42 which depend from and incorporate the limitations of claim 23; claims 33, 39 and 43 which depend from and incorporate the limitations of claim 32; claims 36 and 40 which depend from and incorporate the limitations of claim 21; and claims 37 and 41 which depend from and incorporate the limitations of claim 22. Accordingly, the rejection of claims 24-31 and 33-43 is overcome and should be withdrawn.

As claims 44-47 depend from and incorporate the limitations of claim 1, withdrawal of the Examiner's rejection of these claims is also requested. Furthermore, the elements of claims 44-47 cannot be found in claims 1-5 and therefore, withdrawal of the rejection of these claims is requested on the additional bases discussed above for claims 20-43, namely, lack of a prima facie

case, lack of the claim elements in the cited references, alone or in combination, and for using the same rationale to reject claims 1-5.

2. The Examiner's Rejection of Claims 1-10 and 13-15 under Ji-MacPhail

Claims 1-10, and 13-15 were rejected under 35 U.S.C. §103 as being unpatentable over U.S. Patent No. 5,889,943 to Ji et al (the Ji patent) in view of U.S. Patent No. 4,899,299 to MacPhail (the MacPhail patent).

With respect to claim 1, the Examiner states:

As per claim 1, Ji discloses a method for creating a self-destructing document, comprising the steps of creating an executable module which instructs a computer to automatically delete the document to which the executable module is attached, (when the document based on a preselected expiration date is expired); attaching the executable module to the document [Ji abstract col 3 line 55 - col 4 line 16, col 18 lines 32-54, col 20 lines 30-40]

However Ji fails to detail when the document based on a preselected expiration date is expired. MacPhail discloses a electronic documents is set for automatically delete by an expiration data [MacPhail abstract col 2, lines 48-59]. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to incorporate the message automatically deleted based on an expiration date as taught by MacPhail into Ji's system in order to utilize the email message attach by an executable code would automatically delete by an expiration date. By doing so it would improved [sic] the reliability of data storage on the network.

Again, as claim 6 contains similar limitations to claim 1, except that claim pertains to an e-mail messaging system, the rejection of claim 6 will be discussed together with the rejection of claim 1.

Ji purports to teach an e-mail virus detection and elimination system that polls postal nodes on a computer network for unscanned messages, downloads any found unscanned messages, and performs virus detection and analysis at the node. There is no attached executable module in Ji. Rather, the system works by polling postal nodes. In addition, it discusses the deletion of viruses, not e-mail. Moreover, the deletion is not done in a self-destruct manner, but rather by an entity separate from the virus. Lastly, there is no mention or suggestion of an

expiration date. The e-mails that are analyzed and treated, which is not the same as deleted, are chosen because they are infected with a virus, not because they have expired. This is clear from the abstract alone, which the Examiner has cited for support of his position:

The detection and elimination of viruses on a computer network is disclosed. An apparatus for detecting and eliminating viruses which may be introduced by messages sent through a postal node of a network electronic mail system includes polling and retrieval modules in communication with the postal node to determine the presence of unscanned messages and to download data associated with them to a node for treatment by a virus analysis and treatment module. A method for detecting and eliminating viruses introduced by an electronic mail system includes polling the postal node for unscanned messages, downloading the messages into a memory of a node, and performing virus detection and analysis at the node

Both the apparatus and method of Ji work by polling a postal node, going through each message to pick out those that have not been scanned, and performing virus detection and analysis only. There is no mention or suggestion of creating an attached executable module to instruct a computer to automatically delete a document or e-mail message. It follows, then, that there cannot be any disclosure or suggestion to attach an executable module to a document.

The Examiner also cites col. 3, ln. 55 - col. 4, ln. 16, which states in pertinent part:

Viruses are detected and corrective action taken by a mail scanning apparatus which preferably resides at the client node. The mail scanning apparatus preferably includes: a polling module for determining the presence of unread messages at the postal node, a retrieval module for downloading unread messages to the memory of a client node and a virus analysis and treatment module for determining whether the message contains a virus and for facilitating corrective action to prevent its spread ... The present invention also comprises a method for detecting and eliminating viruses which may spread throughout a network in messages accessed by an electronic mail system. Preferably, the postal node is polled from the client node for unread messages, unread messages are downloaded into the memory of a client node, the messages are scanned for the presence of viruses, and corrective action taken.

With respect to claim 1, the method mentioned in the above paragraph simply includes polling a postal node for unread e-mail, downloading unread e-mail, analyzing the downloaded e-

mail for viruses, and taking corrective action if necessary. There is nothing in the cited passage about a method for creating a self-destructing e-mail by the steps of claim 1. No executable module is created, much less one that instructs a computer to automatically delete a document. Without the creation of an executable module, there cannot be the step of attaching the executable module to the document. Accordingly, the cited passage does nothing to support the Examiner's position.

The modules discussed in the passage are merely parts of the software controlling the system. There is a polling module that polls a postal node for unread messages, a retrieval module that retrieves unread messages, and a treatment/analysis module to analyze e-mails and treat them if infected with a virus. All modules are separate and distinct from the e-mails in the system. None of the modules attach to any e-mails and furthermore, none of the modules delete any e-mails. Therefore, the apparatus taught by Ji cannot even suggest the limitations of claim 6.

In addition, the Examiner has cited col. 18, ln. 32-54 which repeats and reiterates the method of Ji in more detail with respect to a flow chart. The same arguments that apply to the abstract and to the portion from col. 3 and 4 apply here as well.

Lastly, the Examiner cites col. 20, ln. 30-40, which states:

If it is determined 1520 that the attachment cannot be completely cleaned, then it is partially treated 1550 in accordance with the configuration settings. Such treatment may include any of cleaning those portions of the attachment which can be cleaned, deleting the attachment altogether, stripping infected portions from the message, leaving the infected attachment file intact and providing a warning to the recipient. As with the completely treated attachment, the partially treated attachment may be used to replace the infected one in the attachment storage location 300 or may be forwarded or resent to the recipient 1530.

The above portion of Ji is not discussing the deletion, much less the self-destruction, of a document or e-mail message. The "attachment" above is not an attached executable, but rather, is an e-mail attachment, such as a document or image. The portion purports to discuss how an infected attachment (having a virus) would be treated. So where the cited passage states

“deleting the attachment altogether,” it is talking about deleting an infected attachment to an e-mail, not an attached executable deleting a time-expired document or message to which the executable is attached.

MacPhail does not cure the deficiencies of Ji, nor does it add anything to the Examiner’s position. The Examiner looks to MacPhail to teach an electronic document set to automatically delete by an expiration date.

Combining Ji and MacPhail still does not disclose or suggest an executable module attached to a document or e-mail message, the executable module configured to instruct a computer to delete the document or e-mail message to which the module is attached when the document or message expires. Both Ji and MacPhail are polling systems, both work the same way, by polling the contents of a system to find a specific type of data. As noted in applicants’ prior responses of December 13, 2000 (page 11) and July 13, 2000 (page 14), such systems are distinguished in the present specification at page 2, lines 6-19. Combining MacPhail with Ji does not change the way Ji works. The resulting system, according to the Examiner’s logic, takes the polling and deletion mechanism from Ji and the concept of time-expired data from MacPhail, creating a system that polls its contents for time-expired data, instead of polling for viruses, and deleting the time-expired data. This resulting system is simply the same as that which MacPhail claims to recite. Applicants point out to the Examiner that the MacPhail reference has been successfully distinguished in previous amendments on this very basis. An excerpt from a previous response is set out below:

The MacPhail patent purports to disclose a document retention system that establishes a dual label for each document stored in the system with each label having a different expiration date for the document. Retention and deletion selection criteria are accepted from a user at the same time the document is filed by the system. Filed documents are polled by the system to check expiration dates stored in the document labels. If the expiration date has passed, the system deletes the document. An executable module does not delete a document to which the module is attached so there cannot be a self-destructing document. There is no indication in the part of MacPhail referenced by the Examiner (col. 2, line 35-col. 3 line 50), or anywhere in the rest of the MacPhail patent that discloses or suggests creating an executable module at all, much less attaching an executable module to a document.

The MacPhail patent relates to a retention management scheme in which a computer polls each folder and file on a computer, determines the expiration date of the folder or file, and then deletes the file or folder if the current date is later than the expiration date. In this regard, the MacPhail patent is similar to the prior art systems described in the specification of the present application at page 2, lines 6-19.

In view of the above, Ji-MacPhail also fails, at the very least, to disclose an executable module that automatically deletes a message to which the executable module is attached when the message, based upon a preselected expiration date, is expired.

Therefore, for the reasons set out above, it is respectfully submitted that the Examiner's rejection of claims 1 and 6 is overcome and should be withdrawn.

As claims 2-5 and 7-10 and 13-15 depend from and include the limitations of claims 1 and 6, respectively, withdrawal of the Examiner's rejection of those claims is also requested on this basis as well.

3. The Examiner's Rejection of Claims 17-47 under Ji-MacPhail-Shear

Claims 17-47 were rejected under 35 U.S.C. §103 as being unpatentable over Ji in view of MacPhail and further in view of U.S. Patent No. 5,410,598 to Shear (the Shear patent).

With respect to claim 17, the Examiner maintains that

... Ji-MacPhail disclose a self-destructing email messaging system comprising an executable module, the executable module configured to instruct a computer to automatically delete an email message to which the executable module is attached when a predetermined condition is met; an email messaging system, the email messaging system configured to create the message and to transmit the message, the email messaging system attaching the executable module to the message prior to transmission [Ji abstract col4 line 55-col 4 line 16, col 18 lines 32-54, col 20 lines 30-40][MacPhail abstract, col 2 lines 48-59]

However Ji-MacPhail fail to teach said predetermined condition is selected from the group consisting of attempt to print, copy, forward the message. Shear taught this well-known technique in the network security art such as a security

database system with encryption and decryption data including the self-destruction option when user attempt to access an unauthorized feature [Shear abstract, col 18 line 55-col19 line 19]. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to incorporate the message automatically deleted based upon an attempt to access an unauthorized feature as taught by Shear into Ji-MacPhail system in order to prevent the unauthorized data processing on the network.

As discussed above, Ji-MacPhail does not disclose a self-destructing e-mail messaging system as recited in claim 17. Neither reference contains any disclosure or suggestion of creating an executable module at all, much less a module configured to instruct a computer to automatically delete an e-mail message when a predetermined condition is met. Without any disclosure or suggestion of an executable module, it's not possible to teach or suggest attaching an executable module to an e-mail message. The portions cited by the Examiner in the Ji patent are the same as those cited to support his rejection of claims 1-10 and 13-15, and were already distinguished above. The same is true for the cited portions of MacPhail as well. Accordingly, there is no need to address Shear.

In any case, the Examiner relies on Shear to teach the predetermined condition is an attempt to print, copy, or forward the message and maintains that it would be obvious to combine these conditions into the Ji-MacPhail system of automatic deletion.

Shear purports to disclose a data usage metering, billing, and security system that maintains encrypted data and monitors access by measuring the quantity of data that is decrypted. The measured quantity is used to calculate a fee for using the data. Access is limited by usage and is prohibited once the amount of usage expires. The abstract further states "A system may include a 'self-destruct' feature which disables system operation upon occurrence of a predetermined event" The self-destruct feature, however, is not implemented by an executable module attached to the data being destroyed.

In addition, the combination of Ji and MacPhail results in an e-mail system that polls its contents for viruses and expired e-mails. Shear deals with data usage and metering. So combining Ji-MacPhail with Shear results in an e-mail system that polls its contents and keeps

track of data usage. The resultant system does not have a self-destructing e-mail messaging system, comprising an executable module that instructs a computer to automatically delete an e-mail message to which the executable module is attached when a predetermined condition is met, wherein said predetermined condition is selected from the group consisting of an attempt to print the message, an attempt to copy the message and an attempt to forward the message, and an e-mail messaging system that creates the message, transmits the message, and attaches the executable module to the message prior to transmission.

Further more, there is no motivation to combine Ji-MacPhail with Shear. Ji-MacPhail deals with viruses and expiration dates in e-mail systems where Shear is concerned with data usage and metering. The two fields are unrelated as Ji-MacPhail deals with communication and Shear is concerned with database access.

In view of the foregoing reasons, it is respectfully submitted that the rejection of claim 17 is overcome and should be withdrawn.

It is the Examiner's opinion that "Claims 18-47 contain the similar limitations that were addressed in rejecting claims 1-17 above. By the same rationale applied above, claims 18-47 are rejected."

The same arguments set forth above against the Examiner's rejection of claims 18-47 under Thorne-Beck apply here as well. A *prima facie* case has not been made against claims 18 and 19 because the Examiner has failed to point out where their limitations can be found in the Ji-MacPhail-Shear combination. The elements of claims 18 and 19, namely an encrypted document or message and an executable module configured to decrypt the document or message if it is not expired and to delete the document or message if it is expired, cannot be found in Ji-MacPhail-Shear. Further, since the Examiner bases his rejection of claims 18 and 19 on the same rationale as the rejection of claims 1-5, applicants submit that the rejection of claims 18 and 19 are overcome for the same reasons that claims 1-5 are allowable. Therefore, the rejection of claims 18 and 19 are overcome and should be withdrawn.

Applicants reiterate that the Examiner is mistaken in that claims 20-43 "contain the same limitations addressed in rejecting claims 1-17" and that the rejection of claims 20-43 is overcome and should be withdrawn for the same reasons as previously state in regard to the Examiner's rejection of claims 20-43 under Thorne-Beck. These reasons, as set forth above, are the lack of a *prima facie* case because the Examiner has not shown where the limitations of claims 20-43 can be found in Ji, MacPhail, Shear, or any combination of the three. The elements of claims 20-43, namely the methods of creating a virtual container, extracting a virtual container, transmitting a destructible digital object, and a virtual container system, among others, cannot be found in Ji, MacPhail or Shear, alone or in combination.

Claims 44-47 depend from and incorporate the limitations of claim 1, which is believed to be allowable. Accordingly, claims 44-47 are also allowable.

Applicants once again note that the elements of claims 44-47 cannot be found anywhere in claims 1-17. As a result, no grounds for rejecting the limitations of those claims have been provided because the Examiner has not pointed out where the elements of claims 44-47 can be found in the prior art. In any event, even if grounds were provided based on the cited references, claims 44-47 would still be allowable because of their dependency from an allowable base claim.

In view of the above, it is respectfully submitted that the Examiner's rejection of claims 17-47 is overcome and should be withdrawn.

4. The Examiner's rejection of Claims 1-10, 13-15, 17-47 Under Hansen-Beck

Claims 1-10, 13-15, and 17-47 are rejected under 35 U.S.C. § 103 as being unpatentable over an article by Hansen (the Hansen article) in view of Beck.

It is the Examiner's opinion that:

As per claim 1, Hansen discloses a method for creating a self-destructing document, comprising the steps of creating an executable module which instructs

a computer to automatically delete the document to which the executable module is attached when the document, based on a preselected expiration date is expired; attaching the executable module to the document [Hansen, page 28 col 2 lines 4-13]

However Hansen fails to detail the a preselected expiration date is expired. Beck discloses a Email message with attachment automatically deleted by a time limit and encryption and decryption keys [Beck col 7 lines 1-18]. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to combine the technique of a email message automatically deleted by an expiration date as taught by Beck and Hansen's system. By doing so it would improve the security and reliability for message storage and transaction between client/server.

Applicants respectfully submit that Hansen does not disclose a method for creating a self-destructing document as asserted by the Examiner. Hansen purports to discuss enhancing digital documents with embedded functions, for example, embedding a graphics animation into a digital birthday card. Hansen does not mention or even suggest using the embedded functions to implement a method or system for self-destruction of a file, in any form. The technology covered in the article is irrelevant and unrelated to the claims of the application.

In the appendix of Hansen, there is an example of embedding a function to create an extended document. A "Happy Birthday" card includes a picture of an image of a cake and some text. This is the plain, bare document. Using Ness (programming language of Hansen) animation, sound and more text are added. As stated on page 30, column2, line 1, "After empowering the Ness in the birthday card below, the reader can click the mouse on the cake; the card plays 'Happy Birthday', shows the words, and lights the candle on the cake." The examples provided in Hansen do not discuss adding anything other than visual effects, sound effects and text. There is not even a suggestion of adding any type of self-deletion functionality, much less the self-destruct functionality as claimed.

The Examiner cites page 28, column 2, lines 4-13, which states:

Embedding of scripts in documents does not introduce a new level of security problem, but makes more obvious a common security problem. The problem is that in small operating systems when I execute a program written by someone else it may do anything I myself may do; in particular, delete a file, modify a file, or

send a copy of a file—say a forth-coming examination—to an interloper, perhaps a student about to take that examination. Since a Ness script is a program, and since it can do anything a user can, its execution is a security loophole.

In this section, Hansen is merely stating a common problem, and admits to such in the cited portion. The first sentence describes the embedding of scripts (programs) as “a common security problem.” The problem is that when programs written by someone else are executed, the program may perform any function a user may perform. Hansen mentions deleting a file as an undesired function to be prevented and does not discuss deletion as an intended result of an embedded script.

Hansen does not disclose automatically deleting a document, e-mail message, or any data for that matter. As previously discussed, Hansen only eludes to deleting a document and only as a function to prevent, not as an intended result. There is nothing in Hansen about expiration based on time, a date, or any other criteria.

With regard to Beck, it does not “disclose a Email message with attachment automatically deleted by a time limit” as maintained by the Examiner. In the section of Beck cited by the Examiner (col. 7, ln. 1-18), it is clearly stated “... an attachment may in alternative preferred embodiments be automatically deleted after ... a given time limit, such as 90 days.” (Col. 7, ln. 5-9). The attachment is deleted, not the e-mail. The e-mail remains untouched. Furthermore, there is no suggestion that the attachment is deleted by an executable module attached to the attachment or the e-mail.

Since the combination of references cited by the Examiner do not disclose or suggest each of the limitations recited in claims 1 and 6, claims 1 and 6 are believed to be nonobvious and allowable in view of the prior art.

Claims 2-5, depend from and include all the limitations of base claim 1, which has been distinguished above. Accordingly, claims 2-5 should also be allowable based on their dependency therefrom.

Claims 7-16 depend from and include all the limitations of base claim 6, which has been distinguished above. Accordingly, claims 7-16 should also be allowable based on their dependency therefrom.

As the Examiner maintains that claim 17 contains the same limitations as claims 1 and 6, applicants respectfully submit that the rejection of claim 17 is overcome and should be withdrawn for the same reasons as claims 1 and 6.

With respect to claims 18-47, the Examiner has again ignored the different limitations of those claims. Therefore, the same reasons previously set forth for the applicants' request to withdraw the rejection of those claims applies here. The Examiner has not addressed the encryption elements of claims 18 and 19, the methods for creating a virtual container, extracting a document from a virtual container, creating a virtual container and extracting a digital object from a virtual container, transmitting a destructible document of claims 20, 21, 23, and 32, respectively, the virtual container system of claim 22. Furthermore, Hansen-Beck still does not teach or suggest the limitations of claims 18-43.

The Examiner has ignored the different limitations of claim 20-47 and again has not pointed out how the references disclose a method for creating a virtual container, a method for extracting a document from a virtual container, a virtual container system, a method for transmitting a destructible document, or even anything that resembles the claimed virtual container.

In any event, there is simply nothing in Hansen that even remotely suggests the virtual container embodiment of claims 20-43.

Accordingly, the rejection of claims 18-47 is overcome and should be withdrawn.

5. The Examiner's rejection of Claims 1-10, 13-15, 17-47 Under Drake-Norin

Claims 1-10, 13-15, and 17-47 have been rejected under 35 U.S.C. § 103 as being

unpatentable over U.S. Patent No. 6,006,328 to Drake (the Drake patent) in view of U.S. Patent No. 5,787,247 to Norin et al. (the Norin patent).

The Examiner supports his position by stating:

As per claim 1, Drake discloses a method for creating a self-destructing document, comprising the steps of creating an executable module which instructs a computer to automatically delete the document to which the executable module is attached when the document, based on a preselected expiration date is expired; attaching the executable module to the document [such as a message with a header is attached by a executable code or software which is designed to self-destruct, Drake Fig 10, col 7 lines 43-52]

However Drake fails to detail the a preselected expiration date is expired. Norin discloses a Email message with time-based expiration date wherein an object is older a set time will be deleted automatically [Norin col 24 lines 1-25]. Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to combine the technique of a email message automatically delete by an expiration date as taught by Norin and Drake's system. By doing so it would improve the reliability for data storage and transaction between client/server.

Applicants respectfully submit that Drake does not disclose a method for creating a self-destructing document as alleged by the Examiner. Rather, Drake purports to disclose a security system for computer software to prevent certain attacks on executable software by persons or other software. There is no attached executable in the entire reference, so it follows that the remaining limitations of claims 1 and 6 cannot be disclosed in Drake.

The Examiner has cited Fig. 10 of Drake for support. Fig. 10, as stated in Drake at col. 4, ln. 26-27, "illustrates an altered executable form constructed in accordance with the specific embodiment." Fig. 10 depicts a new executable, indicated generally by reference numeral 30. This new executable is the result of some processing by the security system of Drake which takes an old executable and transforms it into a more secure version. Column 14, lines 7-26 summarize the system by stating:

Turning now to Fig. 6, the preferred embodiment of an applicator program 60 is shown which takes as its input the executable program 16 and performs

obfuscating step 61, a ciphering step 62 and an anti-key press and authentication step 63 (described hereafter) which performs various transformations on the executable program 16 to produce a new executable program 30.

The obfuscating step 61 modifies the header 71 (FIG. 7) of the executable 16 in addition to inserting loading code which will be described hereinafter. The cipher step 62 encrypts the existing executable 16 and calculates check data (eg: a checksum) for the encrypted executable. The anti-key press and authentication step 63 replaces various insecure system calls with safe equivalent code and preferably inserts code to graphically represent the integrity of said executable program.

The newly formed executable 30 (new.exe) can be then stored on disk and the applicator program 60 completed, the new executable 30 replacing the old executable program 16.

The new executable is not attached to anything, nor does it delete any data. It merely replaces a pre-existing executable to function in the same way, but with enhanced security. So it is not correct to say that Drake "discloses a method for creating a self-destructing document, comprising the steps of creating an executable module which instructs a computer to automatically delete the document to which the executable module is attached when the document, based on a preselected expiration date is expired; attaching the executable module to the document ... " as stated by the Examiner.

The Examiner cites column 7, lines 43-52, which states:

Using strong cryptographic schemes such as DES "(Data Encryption Standard)", or RSA "(RIVEST, SHAMIR, ADELMAN, Algorithm)" or the like will present the examination of any decryption routines from revealing a simple patch to disable said routines. When tracing software, the program stack is usually used by the debugger either during the tracing operations or at other times. This is easily detected, and by using the area of the stack which will be destroyed by unexpected stack-use for code or critical data, software can be designed to self-destruct in this situation.

The above paragraph discusses using a certain area of computer memory that is deleted whenever a specific type of unexpected use of memory occurs. By intentionally storing specific information in this particular area, the information is purportedly deleted when the unexpected use of memory occurs ("by using the area of the stack which will be destroyed by unexpected stack-use for code or critical data, software can be designed to self-destruct in this situation").

This is not a self-destruct function, deletion is done by the system of certain areas of memory. There is no attached executable involved, and furthermore, e-mail is not even mentioned in the reference.

The Examiner combines Drake with Norin for its discussion of time-expiration data. However, the time-expiring e-mails discussed in Norin involve a polling system, which has already been distinguished. Therefore, Norin adds nothing to Drake so combining the two references still does not teach or suggest the limitations of claims 1 and 6.

It is therefore respectfully submitted that the rejection of claims 1 and 6 have been overcome and should be withdrawn.

As claims 2-5, 18 and 44-47 depend from and include all the limitations of claim 1, withdrawal of the rejections of these claims is also respectfully requested.

As claims 7-16 and 19 depend from and include all the limitations of claim 6, withdrawal of the rejections of these claims is also respectfully requested.

The Examiner still has not made a *prima facie* case of obviousness for claims 20-47 with the Drake-Norin combination. He has not indicated where the elements of claims 20-47 can be found in either reference singularly, or in combination. The Examiner has rested his rejection of claims 20-47 on the “same rationale applied above” for rejecting claims 1-5. Claims 20-47, however, do not share any similarities with claims 1-5, as previously discussed.

It cannot even be said that Drake, or Norin, contain any of the limitations of claims 20-43 which recite the virtual container embodiment of the present invention. There is nothing, anywhere, in either reference, that even resembles the method for creating a virtual container of claim 20, the method for extracting a document from a virtual container of claim 21, the virtual container system of claim 22, the method for creating a virtual container and extracting a digital object from a virtual container of claim 23, or the method for transmitting a destructible digital object of claim 32. It logically follows, therefore, that there is also nothing in either reference

that suggests the limitations of their dependent claims.

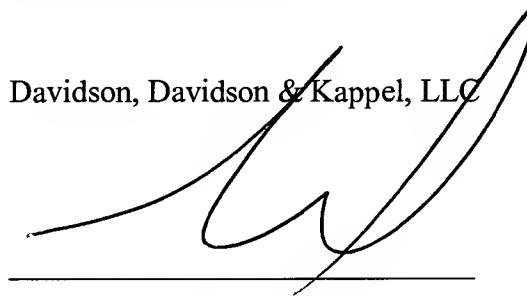
In view of the above, applicants respectfully submit that the Examiner's rejection of claims 20-47 is overcome and should be withdrawn.

The present invention is new, useful, and unobvious. Reconsideration and allowance of the present application is therefore respectfully requested.

Respectfully submitted,

Davidson, Davidson & Kappel, LLC

By :


Cary S. Kappel (Reg. 36,561)

485 Seventh Avenue, 14th Floor

New York, NY 10018

212-736-1940 (phone)

212-736-2427 (fax)